Department of Pesticide Regulation

Paul E. Helliker
Director

Gray Davis
Governor

Winston H. Hickox
Secretary, California Environmental Protection Agency

MEMORANDUM

TO: Randy Segawa
Senior Environmental Research Scientist
Environmental Monitoring Branch

FROM: LinYing Li
Associate Environmental Research Scientist
Environmental Monitoring Branch
(916) 324-4118

DATE: January 10, 2002

SUBJECT: WEATHER ANALYSIS FOR MONTEREY, SANTA CRUZ, AND KERN COUNTIES DURING THE AIR RESOURCES BOARD AMBIENT AIR MONITORING PERIOD FOR METHYL BROMIDE IN YEAR 2000

In order to evaluate the weather conditions during the Air Resources Board (ARB) ambient air monitoring period for methyl bromide in year 2000, weather data from three CIMIS weather stations in Monterey, Santa Cruz, and Kern Counties were analyzed using statistical methods. These weather stations are: station 116 (Salinas-North), station 129 (Pajaro) and station 125 (Arvin-Edison). The key weather elements affecting the air dispersion of methyl bromide were compared for the monitoring periods between the monitoring year (2000) and the mean of the previous five years. The focus was on solar radiation, air temperature, wind speed, and direction, and atmospheric stability. It was concluded that the weather conditions in year 2000 during the ARB ambient air monitoring period for methyl bromide were in the normal range compared to the local weather conditions in historical years. Weather conditions in Kern County were more suitable for air dispersion than that in Monterey, and Santa Cruz Counties. For detailed information about this analysis, please read the attached report, "Weather Analysis for Monterey, Santa Cruz, and Kern Counties during ARB Ambient Air Monitoring Period for Methyl Bromide in year 2000."

Weather analysis for Monterey, Santa Cruz and Kern Counties during ARB ambient air monitoring period for methyl bromide in year 2000

LinYing Li

Associate Environmental Research Scientist
Department of Pesticide Regulation
Environmental Monitoring Branch

1. Introduction

The Air Resources Board (ARB) monitored the ambient air concentration of methyl bromide during peak use periods in Monterey/Santa Cruz Counties and Kern County in year 2000[1,2]. The monitoring period lasted 8 weeks in Monterey/Santa Cruz Counties (09/11/00 ~ 11/02/00), and 7 weeks in Kern County(07/19/00 ~ 08/31/00). The Department of Pesticide Regulation (DPR) analyzed the data and found that air concentration was significantly correlated to the use of methyl bromide over various areas with the monitoring site as centroid [3]. The best fit using a linear regression model occurred between air concentration and use in a 7x7 mile^2 area over a 7~8 week period (the R^2 was 0.95), and was recommended as a basis for calculating use limits to mitigate subchronic exposure. However, meteorological factors were not included in the above analysis. Weather condition is one of the most important factors affecting spatial and temporal distributions of air pollutants, especially for short term air dispersion. Therefore, weather patterns should be examined to determine whether or not weather conditions were abnormal during the monitoring period. This report documents the methods, procedures and results of weather pattern analysis for Monterey, Santa Cruz and Kern Counties during ARB ambient air monitoring period for methyl bromide in year 2000.

2. Materials and Methods

2.1 Weather stations

The California Irrigation Management Information System (CIMIS) has many weather stations in Monterey/Santa Cruz Counties and Kern County (Fig.1). The following criteria were used when considering which weather station(s) to use for this analysis: the station 1) must be in Santa Cruz, Monterey or Kern Counties and relatively close to one or more monitoring sites; 2) must be currently operating; and 3) must have records 5 years or longer. Eleven weather stations were selected from 32 possible stations in Monterey/Santa Cruz Counties and Kern County according to these criteria (Table 1).

Since there are multiple weather stations and multiple monitoring sites, a number of questions arise with regard to selecting the 'best' weather station: 1) which weather station should be used for a particular monitoring site? 2) if only one weather station can

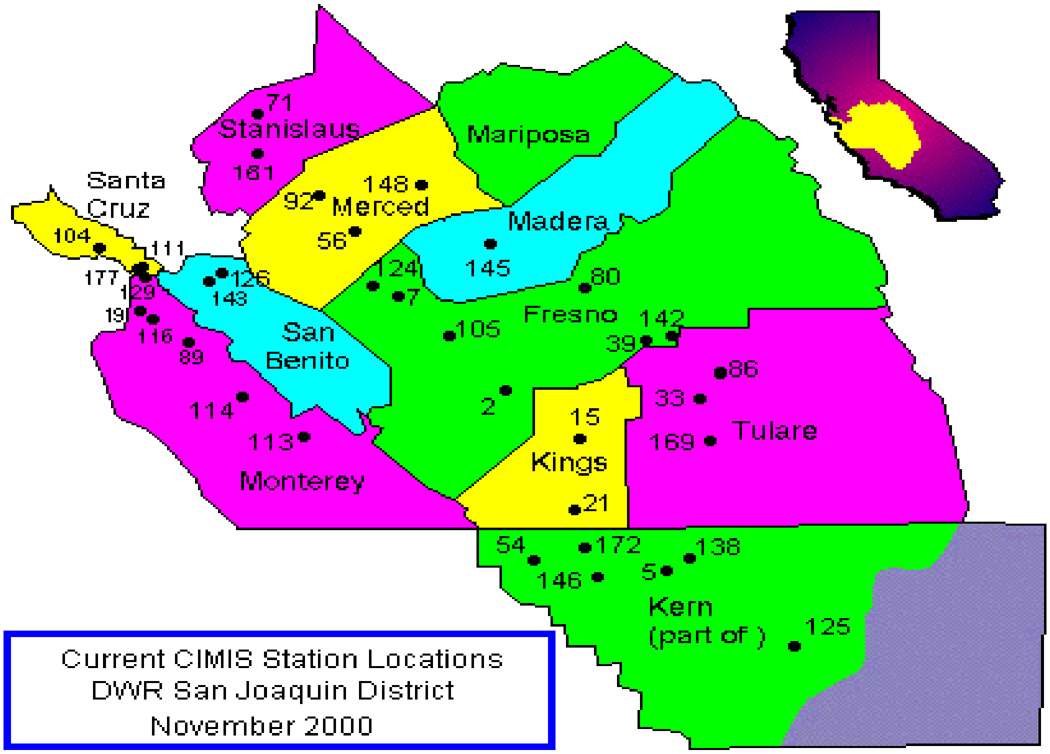

Figure 1 CIMIS weather stations in San Joaquin District

Table 1. CIMIS weather stations in Santa Cruz, Monterey and Kern Counties (currently running, with 5 years or longer records and close to at least one of the monitoring sites)

Station	Station Name	Nearby City	County	Starting date	Years of records	Long	Lat	Elev (ft)
104	De Laveaga	Santa Cruz	Santa Cruz	09/28/90	10	121.996	36.998	300
111	Green Valley Road	Watsonville	Santa Cruz	05/29/92	8	121.767	36.941	110
19	Castroville	Castroville	Monterey	11/18/82	18	121.775	36.768	9
89	Salinas South	Salinas	Monterey	09/05/92	8	121.528	36.525	120
113	King City-Oasis Rd.	King City	Monterey	06/12/93	7	121.084	36.121	540
114	Arroyo Seco	Soledad	Monterey	06/18/93	7	121.290	36.359	235
116	Salinas North	Salinas	Monterey	06/18/93	7	121.691	36.717	61
129	Pajaro	Pajaro	Monterey	09/13/95	5	121.742	36.903	65
5	Shafter/USDA	Shafter	Kern	06/01/82	18	119.281	35.533	360
54	Blackwells Corner	Blackwells Corner	Kern	10/19/86	14	119.958	35.650	705
125	Arvin-Edison	Arvin	Kern	03/22/95	5	118.778	35.206	500

be used for all of the monitoring sites, which one is the most representative? 3) how well is a monitoring site represented by a set of weather stations? A simple measurement of representativeness might be the distance between a weather station and a monitoring site with the understanding that closer is better. Therefore, the Euclidian distances were calculated between monitoring sites and weather stations based on their longitudes and latitudes (Table 1 and 2). The following equation was used to calculate the Euclidian Distance when the two sites are not far apart from each other:

$$D = 2\pi R/360\sqrt{[Lat_1 - Lat_2]^2 + [\frac{1}{2}(Long_1 - Long_2)(\cos(Lat_1) + \cos(Lat_2))]^2} \quad (1)$$

where $Long_1$, $Long_2$ and Lat_1, Lat_2 represent the longitudes and latitudes of weather station and monitoring site. R is the radius of the earth (6370 km), 2π/360 is a factor to convert degrees to radians. This equation calculates the straight line distance between two points, which can be used to approximate the curve distance on the earth surface when these two points are relatively close to each other.

Table 2. GPS coordinates of ARB monitoring sites in Monterey and Santa Cruz Counties

Sites	**Latitude**	**Longitude**
SES	36°57.67'	121°43.88'
PMS	36°53.91'	121°43.95'
LJE	36°43.97'	121°38.05'
SAL	36°41.63'	121°37.39'
CHU	36°34.37'	121°31.00'
OAS	36°18.92'	121°15.10'

Table 3. Euclidian distance (km) between monitoring sites and weather stations in Monterey/Santa Cruz Counties

Station	**SES**	**PMS**	**LJE**	**SAI**	**CHU**	**OAS**	**Average**	**Total**
111	3.9	5.6	26.4	30.8	46.7	83.0	32.7	196.4
113	109.7	104.1	83.5	79.4	63.4	26.7	77.8	466.8
114	77.5	71.8	51.2	47.2	31.2	5.6	47.4	284.5
116	27.2	20.6	5.6	7.0	22.4	59.2	23.7	142.0
129	6.4	1.1	21.7	26.1	42.0	78.3	29.3	175.6
19	21.7	15.2	13.5	16.2	31.6	68.3	27.8	166.5
89	51.6	45.4	24.5	20.1	5.1	33.7	30.1	180.4
Average	42.6	37.7	32.3	32.4	34.6	50.7	38.4	
Total	298.0	263.8	226.4	226.8	242.4	354.8		1612.2

The shortest distance from a weather station to a monitoring site is 1.1 km, which is between Station 129 (Pajaro) and PMS. For each monitoring site, there is at least one weather station within 7 km distance. From Table 3, one can find the closest monitoring site to each weather station, or the nearest weather station for each monitoring site. The column average or column total represents how well weather conditions of a particular monitoring site are represented by this set of weather stations. And the row average or row total represents how well a particular weather station represents the overall weather conditions of all the monitoring sites. The best represented monitoring sites are LJE and

SAL, with an average distance of 32.4 km to each weather station, followed by CHU and PMS. The most representative weather station is station 116 (Salinas North), with an average distance of 23.7 km to each monitoring site. In this analysis, station 116 (Salinas North) and station 129 (Pajaro) were used for their representativeness in Monterey/Santa Cruz counties. On average, station 116 (Salinas North) is the closest weather station to all monitoring sites, and station 129 (Pajaro) is within 1.1 km distance to the PMS monitoring site, where the highest air concentration of methyl bromide was observed. Other factors, such as the distance from the ocean and the elevation, may be important when considering if a weather station is representative to the weather conditions of a monitoring site. However, the stations are reasonably uniform in this regard and the primary focus of this work will be the distance.

Because we do not have GPS coordinates of monitoring sites in Kern County, the Euclidian distances from a particular weather station to monitoring sites are not available in Kern County. From the monitoring site map (Fig. 1 of [3]) and the weather station location map (Fig.1), weather stations 5 and 125 seem to be most representative. However, some critical weather elements, such as wind speed and direction, were missing for station 5. The only station that can be used was station 125 (Arvin-Edison) in Kern County.

The Euclidian distance was calculated with a computer program (Appendix-I: A perl program to calculate the Euclidian distance between two sets of locations)

2.2 Weather elements

Hourly averages of various meteorological elements can be retrieved from CIMIS database. CIMIS hourly data records contain 17 fields, some are directly measured and others derived from measured values. Fields 6 and 7 are reserved for experimental purposes, and field 13 is not used. Weather conditions can be characterized by many meteorological elements. However, not all of these elements exert the same degree of influence on the dispersion of methyl bromide in the air. The following elements were the focus when comparing the weather pattern of the monitoring year to the historical years:

(1) solar radiation
(2) air temperature
(3) wind speed
(4) wind direction
(5) atmospheric stability

Solar radiation is a major driving force of atmospheric activities near the ground, and particularly vertical convective mixing. Air temperature is a measurement of thermal status of atmosphere. Its vertical gradient near ground surface dictates the intensity of free convection, such as molecular diffusion and small eddy movement. Atmospheric stability is a more direct measurement of vertical mixing intensity of low atmosphere. Wind is a driving force of air mixing in a large scale, wind speed and direction determine the effectiveness and spatial distribution of dissipation.

2.3 Data Analysis

2.3.1 Mean, standard error of mean and the confidence interval of mean

For each meteorological element, the average was calculated for each year from 1995 to 2000. All averages were over the same time period as the monitoring period. The mean of the averages(grand mean, or second order mean) was calculated from the data of the first 5 years (Year 1995 to Year 1999) and its 95% confidence interval was then calculated and compared to the average of the monitoring year (Year 2000). If the average of the monitoring year was within the 95% confidence interval of the population mean estimated from the previous 5 years, then the meteorological element in monitoring year was not significantly different from the mean, and was considered a normal status. The same method also applies to other indices derived from meteorological elements, such as frequency distributions of wind speed and wind direction and stability classes.

The mean is calculated as the arithmetic average:

$$\overline{Y} = \frac{\sum_{i=1}^{n} Y_i}{n} \tag{2}$$

where Y_i is a meteorological variable calculated over the sampling period of each year, and n is the number of years.

The sample standard deviation is defined as

$$\hat{\sigma} = \sqrt{\frac{\sum (Y_i - \overline{Y})^2}{n-1}} \tag{3}$$

The standard error of the mean (SEM) is then expressed as

$$\sigma_{\overline{Y}} = \frac{\hat{\sigma}}{\sqrt{n}} \tag{4}$$

which measures the dispersion of sampling distribution, and is thus an index of variability of $\overline{Y}$ from sample to sample.

The confidence interval of mean was calculated as:

$$CI = \overline{Y} \pm t_{\alpha/2}(\frac{\hat{\sigma}}{\sqrt{n}}) \tag{5}$$

where $t_{\alpha/2}$ is the t value with a degree of freedom of n-1. For the 95% significant level ($\alpha = 0.05$) and n = 5, $t_{\alpha/2}$ = 2.776. For large number sampling, the error bar of using the 95% confidence interval is approximately twice as wide as using the standard error of mean ($t_{\alpha/2}$ = 1.96 when n >=30) [4].

The calculation of mean, standard deviation and confidence intervals was quite different for wind speed and wind direction, which are considered as variables with circular distributions[5]. Here we only introduce the method of calculating average wind speed and wind direction.

2.3.2 The average of vector wind speed and direction

Wind speed is a vector variable, with both scalar and directional components. The method described in the above section can only be used for scalar values. To calculate the average wind speed and direction, the vector wind speed must be cast on the X-axis and Y-axis and then obtain the averages of x components and y components of vector wind speeds[6] :

$$\overline{x} = \sum \frac{S_i \sin(\theta_i)}{n} \tag{6a}$$

$$\overline{y} = \sum \frac{S_i \cos(\theta_i)}{n} \tag{6b}$$

where S_i and θ_i represent respectively the scalar wind speed and direction of one observation.

The average wind speed $\overline{U}$ is calculated as:

$$\overline{U} = \sqrt{\overline{x}^2 + \overline{y}^2} \tag{7}$$

and the average wind direction is

$$\overline{\theta} = \arctan(\frac{\overline{x}}{\overline{y}}) \tag{8}$$

The average of scalar wind speed was computed as:

$$\overline{S} = \sum \frac{S_i}{n} \tag{9}$$

2.3.3 Frequency distribution of scalar wind speed categories

The scalar wind speed was classified into eight categories: 0~1, 1~2, ..., and 7~8 m/s. The frequency of wind speed in these categories was calculated based on hourly wind speed.

2.3.4 Frequency distribution of wind directions and prevailing wind directions

Eight wind direction zones were used: 0^o~45^o,45^o~90^o ,......, 215^o~360^o. The frequency of wind direction in these eight 45^o intervals was computed. However, if the prevailing wind direction is defined as a 45^o interval where the maximum frequency occurred, it is not likely that this 45^o interval happens to be one of the eight conventional wind directional zones. For this reason, the wind direction frequency was calculated on all possible consecutive 45^o intervals: 0^o~45^o,1^o~46^o ,......, 314^o~359^o and 315^o~360^o [6]. The maximum frequency and the corresponding directional interval was recorded.

2.3.5 Stability classification

There are many methods for the stability classification. The Pasquill-Gifford stability classification system categorizes stability into six classes, from A to F. Class A is the most unstable condition and F represents the most stable condition. The Sigma_A method was used in this analysis. This method uses the standard deviation of horizontal wind direction as a criterion for an initial estimate of stability, and then modified by the wind speed at 10 m height on a set of rules separating daytime versus nighttime[7].

Under certain assumptions (neutral atmospheric stability, open and smooth site), the wind profile can be characterized by the logarithmic function [8]:

$$u(z) = \frac{u_*}{k} \ln \frac{z}{z_o} \tag{10}$$

where u(z) is the wind speed at the height z. The variable u_* is the friction velocity, z_o is the roughness of surface, and k is Karman's constant (0.38). The wind speed at any height can be estimated if wind speed is available at a certain height:

$$u_2 = \frac{\ln z_2 - \ln z_o}{\ln z_1 - \ln z_o} u_1 \tag{11}$$

here u_2 and u_1 are wind speeds at heights z_2 and z_1. Utilizing equation (10), the 10-meter height wind speed can be estimated from the CIMIS standard 2-meter wind speed measurement.

The determination of day and night time was based on net radiation. It was nighttime for negative net radiation, otherwise it was defined as daytime. Theoretically, the solar radiation is a better criterion for this classification. However, many positive values were found for solar radiation during nighttime in CIMIS data.

The EPA air dispersion modeling guidance does not allow the stability to change more than one stability class between adjacent hours. The stability class determined by the above rules does not guarantee this, especially during the transition hours between day and night. An algorithm was used to smooth the stability so that it wouldn't change by more than 1 class from hour to hour.

Data analysis was performed by a computer program (Appendix-II: A perl program for CIMIS weather data analysis), which processed the raw data and data for statistical analysis and plotting.

3. Results

3.1 Overall weather conditions

The mean and its 95% confidence interval(CI) estimated from the previous 5 years were compared to the average of the monitoring year for all of 17 meteorological elements (except wind direction and sensors not used or for experiment) in the CIMIS data (Fig 2a, Fig 2b and Fig 2c). Because values of different elements vary in a large range, logarithmic scale was used for the Y-axis. The error bar represented the 95% confidence interval of the mean (symbolized by a small circle in the center of the error bar). The average of the monitoring year over the monitoring period was represented by the square. In general, meteorological elements during the monitoring year were not significantly different from the population mean estimated from the previous 5 years. Some elements vary in a wider range than others. The weather at Station 129 (Pajaro) and Station 116 (Salinas North) showed a very similar pattern, with the same level of means and variation for most elements, such as radiation, temperatures, and wind direction. However, the weather in Kern County (Station 125, Arvin-Edison) was quite different from that of Monterey. Kern weather was warmer and dryer, more solar radiation and less wind speed compared to the weather in Monterey. Monitoring was conducted in summer in Kern County, in the fall in Monterey and Santa Cruz Counties. Thus seasonal and geographical factors (coastal vs inland) contributed to the difference in weather conditions between these two areas.

3.1 Solar radiation and temperature

In Monterey County, the radiation condition was almost the same in Pajaro (S129) and in Salinas North (S116). At both weather stations, the solar radiation was below 110 $Wm^{-2}day^{-1}$ and the net radiation was around 80 $Wm^{-2}day^{-1}$. In Kern County, the solar radiation was about 10% stronger, 120 $Wm^{-2}day^{-1}$ and 110 $Wm^{-2}day^{-1}$ respectively for solar radiation and net radiation . Air temperature and soil temperature were about 1 C^{o} higher at Arvin-Edison(S125) in Kern County than at Pajaro (S129) and Salinas North (S116) in Monterey County. Both radiation and temperatures in the monitoring year were in the normal range derived from the pervious five years. Weather condition was more favorable to the dissipation of methyl bromide in Kern County than in Monterey County.

3.3 Wind speed and wind direction

3.3.1 Average wind speed and wind direction

The average wind speeds and wind directions during the monitoring period from 1995 to 2000 were plotted on a polar coordinate system, in which wind direction was represented by that from the symbol to the center, and the wind speed represented by the length from the symbol to the center (Fig 3a-Fig 3c). The points representing wind speeds and directions are closed clustered, indicating the monitoring year seems not significantly different from the previous years. The average wind speed (weighted by wind direction) was around 0.75 m/s, with a slightly stronger wind at Station 116(Salinas North). The average wind direction at Station 129 (Pajaro) and Station 125(Arvin-Edison) is Southeast (Fig 3a&c), and Northeast at Station 116(Salinas North) (Fig 3b). There is one year's record at Station 129 (Pajaro) which is far away from the cluster, but the monitoring year can considered as the part of the cluster (Fig 3a).

3.3.2 Frequency distribution of wind speed

Stations 129 (Pajaro) and 116 (Salinas North) shared the same pattern of wind speed distribution (Fig 4). The frequency of wind speed category was not statistically different for both stations. Wind speed was 1-2 ms^{-1} approximately 40% of the time, and less than 2 ms^{-1} wind speed counted about 60% of the time. At Station 116 (Salinas North), the frequency of wind speed between 2 to 3 ms^{-1}during the monitoring period was higher than the average for the same period of previous 5 years. The wind speed in Kern County was much smaller, and less than 2 ms^{-1} about 80% of the time.

3.3.3 Frequency distribution of wind direction

Southwest wind occurred more often at Station 129 (Pajaro) and Station 125 (Arvin-Edison) (Fig 5). Because of the large error bar, this dominance was not persistent from year to year. At Station 116 (Salinas North), northwest wind was most frequent and the frequency for northwest wind did not change much from year to year. The wind direction frequency distribution during the monitoring year was not statistically different from historical years.

There was an active fumigation facility to the east of the PMS monitoring site during the monitoring period. Because of the northwest prevailing wind direction, and because the accumulative frequency of wind direction between 45^{o} and 135^{o} was less than 25% (Fig. 5), this fumigation facility should not significantly influence the monitoring results. The high air concentration at PMS site was likely induced by heavy use of methyl bromide for soil fumigation[3].

3.3.4 Prevailing wind direction

The highest frequency of wind direction in a 45 degree interval was about 1/3 at all three weather stations(Table 4). At station 116 (Salinas North), the frequency of prevailing wind direction had a wider confidence interval than at other two stations. The prevailing wind direction was southwest at stations 129 (Pajaro) and 125 (Arvin-Edison). The prevailing wind direction at station 116 (Salinas North) was northwest, and did not change much from year to year. The prevailing wind direction and frequency of the monitoring year were not statistically different from their means.

Table 4. Prevailing wind directions and frequencies of historical years (1995-1999) and the monitoring year

Station	Prevailing wind direction (degree)		Frequency		
	Mean	Monitoring year	Mean	CI	Monitoring year
129 (Pajaro)	205~250	208~253	0.315	0.315 ± 0.141	0.318
116 (Salinas North)	278~323	280~325	0.332	0.332± 0.093	0.333
125 (Arvin-Edison)	212~257	228~273	0.342	0.342± 0.080	0.303

3.4 Stability

At all three stations, the F stability class showed the highest percentage. The F class of stability at the Station 129 (Pajaro) occurred about 40%, and it was significantly higher than the frequency of the same stability class at stations 116 (Salinas North) and 125 (Arvin-Edison) (Fig 6). This high percentage of F stability class at station 129 (Pajaro) was consistent throughout all 6 years. In the monitoring year, the frequency of F class at Station 129 (Pajaro) was slightly lower than the average of previous five years, but the difference was not statistically significant. The A stability class frequency at Station 125 (Arvin-Edison) was much higher than at the station 129 (Pajaro) and station 116 (Salinas North). Therefore, the unstable atmospheric condition in Kern County during the monitoring period was more helpful to the dissipation of methyl bromide in the air.

The stability frequency distribution at stations 129 (Pajaro) and 116 (Salinas North) show the high percentage of F class and very low percentage of unstable classes, a condition that helps the build up of high concentration. Station 125 (Arvin-Edison) had higher frequency for both stable and unstable classes. This reflects some differences in stability characteristics between the costal climate and the inland climate. The latter tends to have bigger temperature difference between day and night, leading to both extremes of the stability condition.

4. Summary

With just a few exceptions, the meteorological elements and other atmospheric stability factors such as wind speeds, wind directions and stability classes during the monitoring period were in the normal range. Although there were higher frequencies of stable atmospheric conditions in Monterey/ Santa Cruz Counties than in Kern County, weather conditions during the monitoring period was not significantly different from normal local weather conditions in previous years. The subchronic air concentrations observed during the ARB ambient air monitoring periods for methyl bromide in year 2000 was measured under normal weather conditions for those areas at that time of the year.

Acknowledgment

This report was reviewed by Bruce Johnson, Terri Barry and Randy Segawa, senior environmental research scientists in the Department of Pesticide Regulation. They offered valuable comments, critics and suggestions which greatly improved this weather analysis report.

Literature Cited

[1] ARB, 2001. Ambient air monitoring for methyl bromide and 1,3-Dichloropropene in Monterey/Santa Cruz Counties - Fall 2000. California Air Resources Board. Sacramento, CA.

[2] ARB, 2001. Ambient air monitoring for methyl bromide and 1,3-Dichloropropene in Kern County - Summer 2000. California Air Resources Board. Sacramento, CA.

[3] Li, LY, B. Johnson and R. Segawa. 2001. Empirical relationship between use, area, and ambient air concentration of methyl bromide for subchronic exposure concerns. Department of Pesticide Regulation, Environmental Monitoring Branch, Sacramento, CA 95812.

[4] Agresti Finlay, 1986. Statistical Methods for the Social Sciences, 2nd Edition, Dellen Macmillan. P253-273.

[5] Zar, Jerrold H., 1996. Biostatistical analysis. 3rd Edition. Prentice Hall.

[6] Johnson, B. R. 1998. Analysis of weather patterns in Lompoc, California. State of California, Environmental Protection Agency. Department of Pesticide Regulation. Sacramento, CA 95814.

[7] US EPA, 2000. Meteorological Monitoring Guidance for Regulatory Modeling Applications. US EPA, Office of Air Quality Planning and Standards, Research Triangle Park, NC 27711.

[8] Rosenberg, N. J., B.L. Blad and S.B. Verma, 1983. Microclimate: The biological environment (2nd Edition). John Wiley & Sons, Inc.

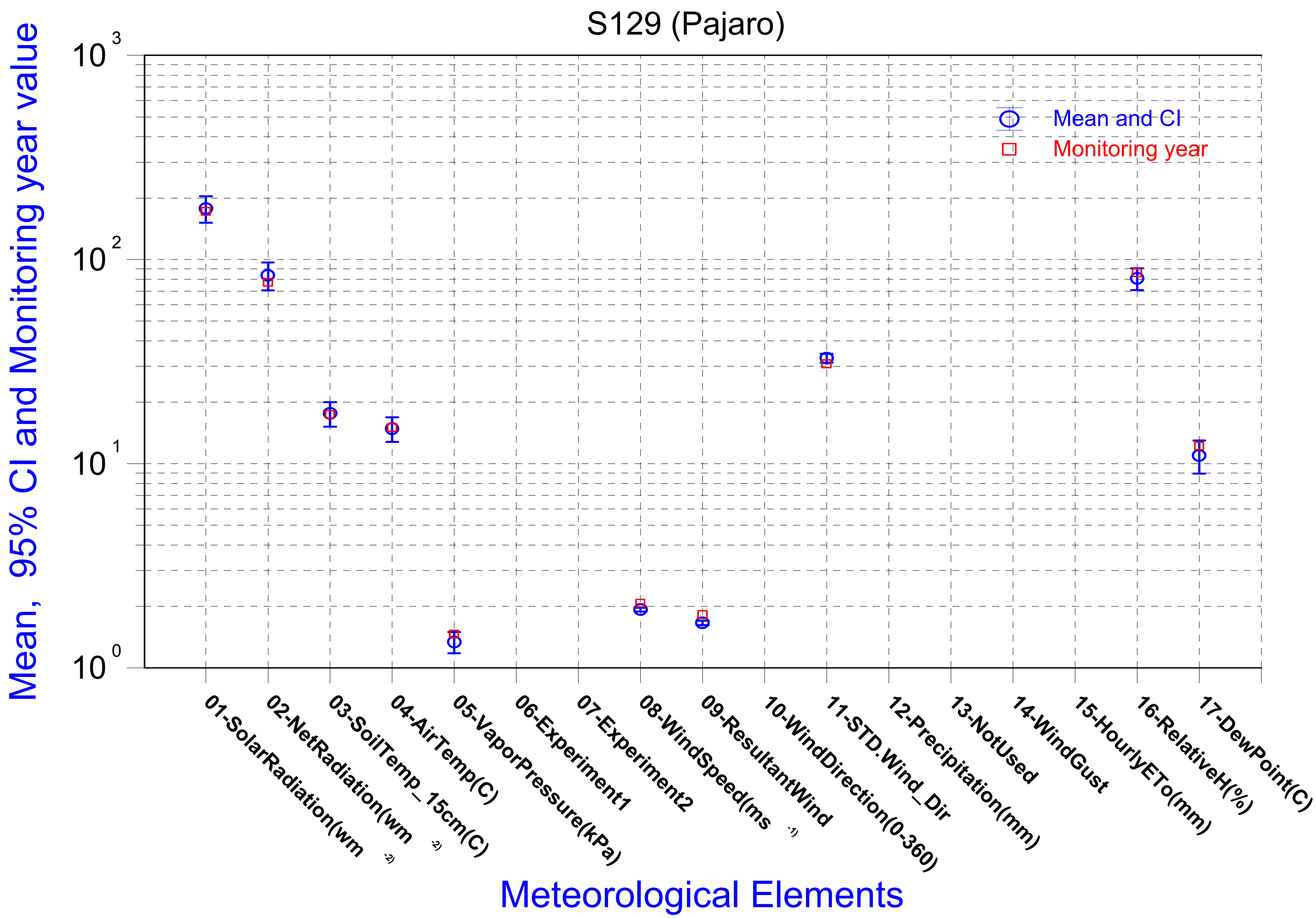

Figure 2a. Mean, 95% confidence interval and monitoring year value of meteorological elements for CIMIS weather station 129

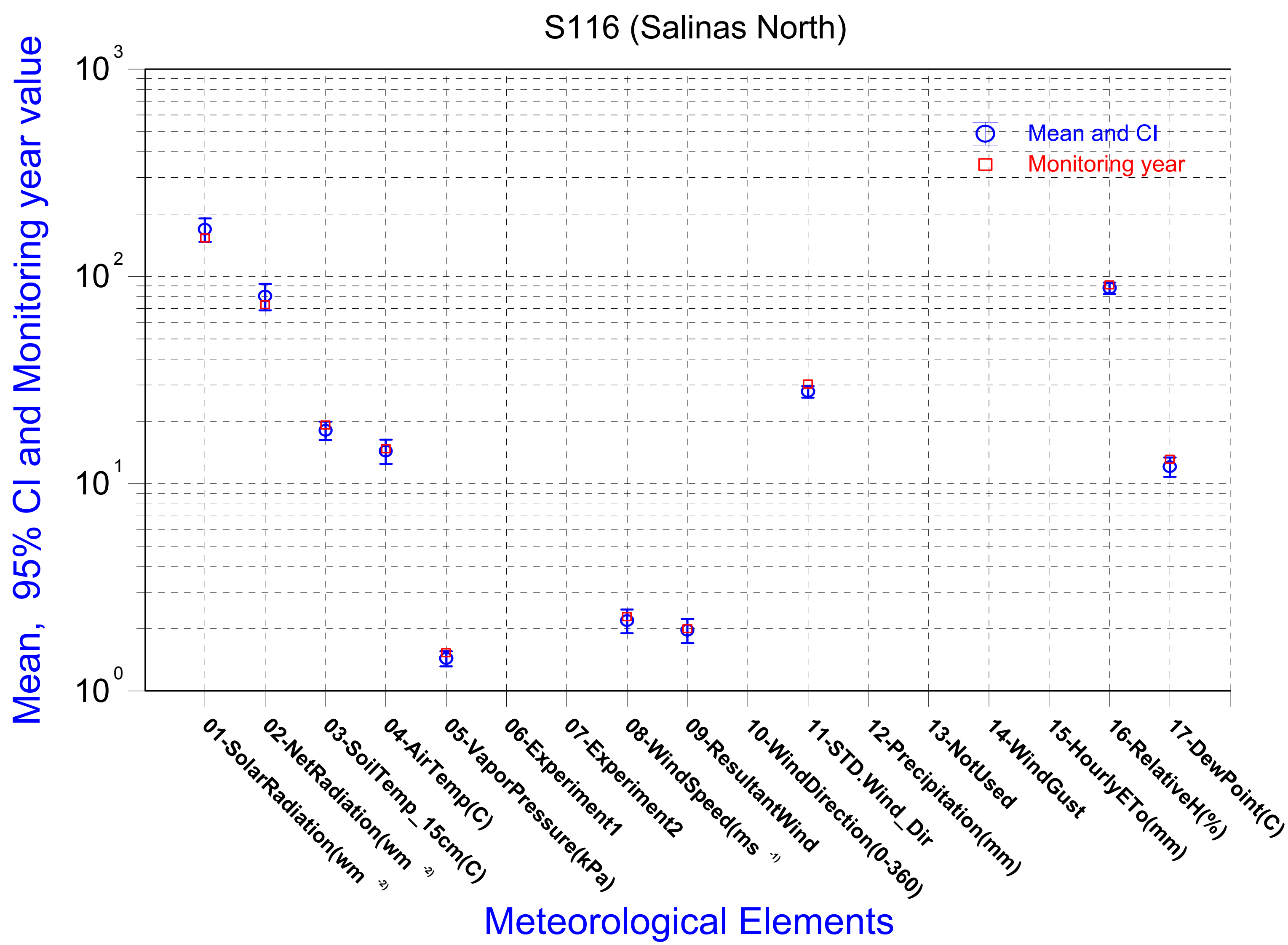

Figure 2b. Mean, 95% confidence interval and monitoring year value of meteorological elements for CIMIS weather station 116

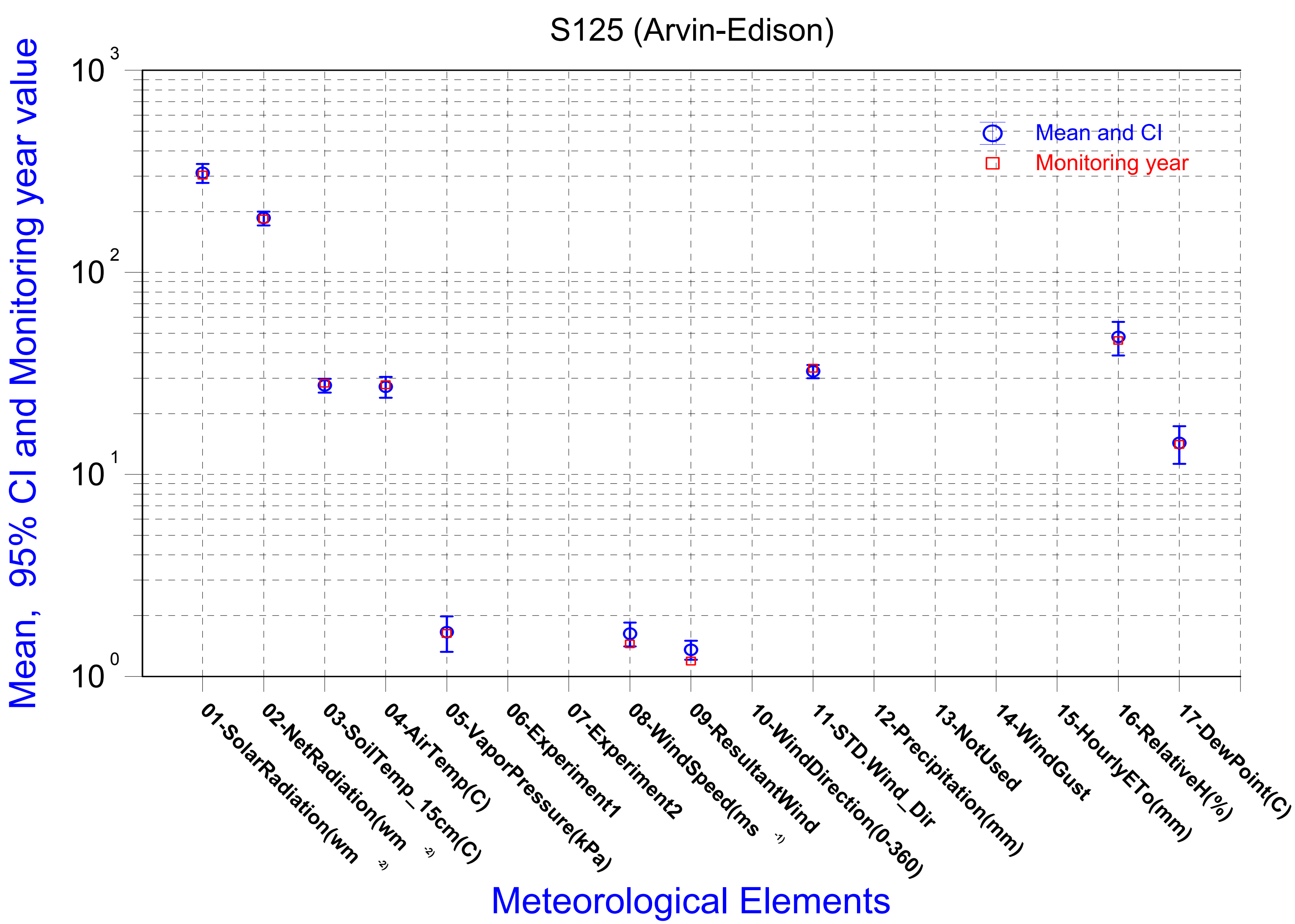

Figure 2c. Mean, 95% confidence interval and monitoring year value of meteorological elements for CIMIS weather station 125

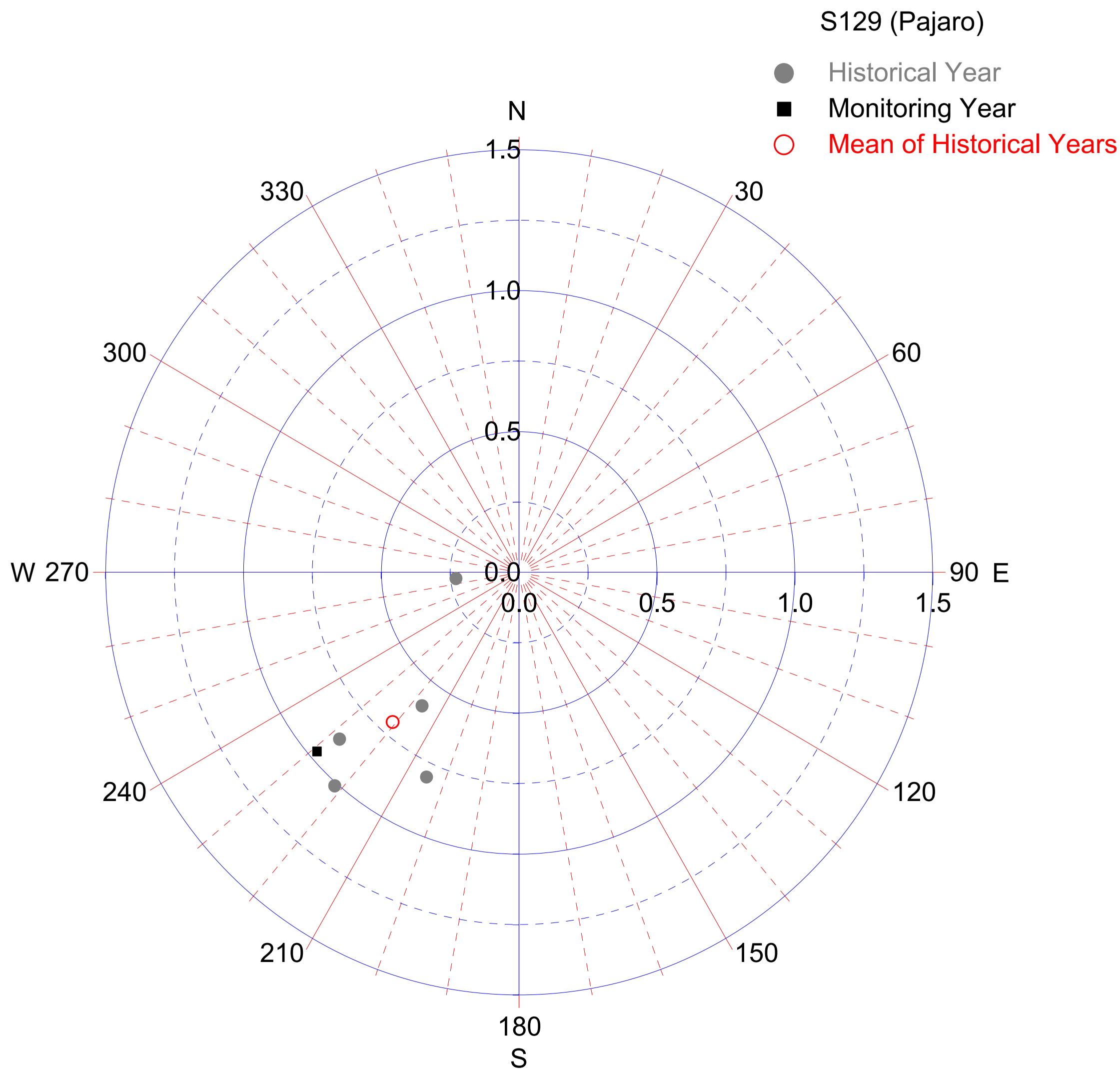

Figure 3a The average wind speed and direction over the monitoring period (S129). The wind direction was from the symbol to the center, and the wind speed was represented by the distance from the symbol to the center

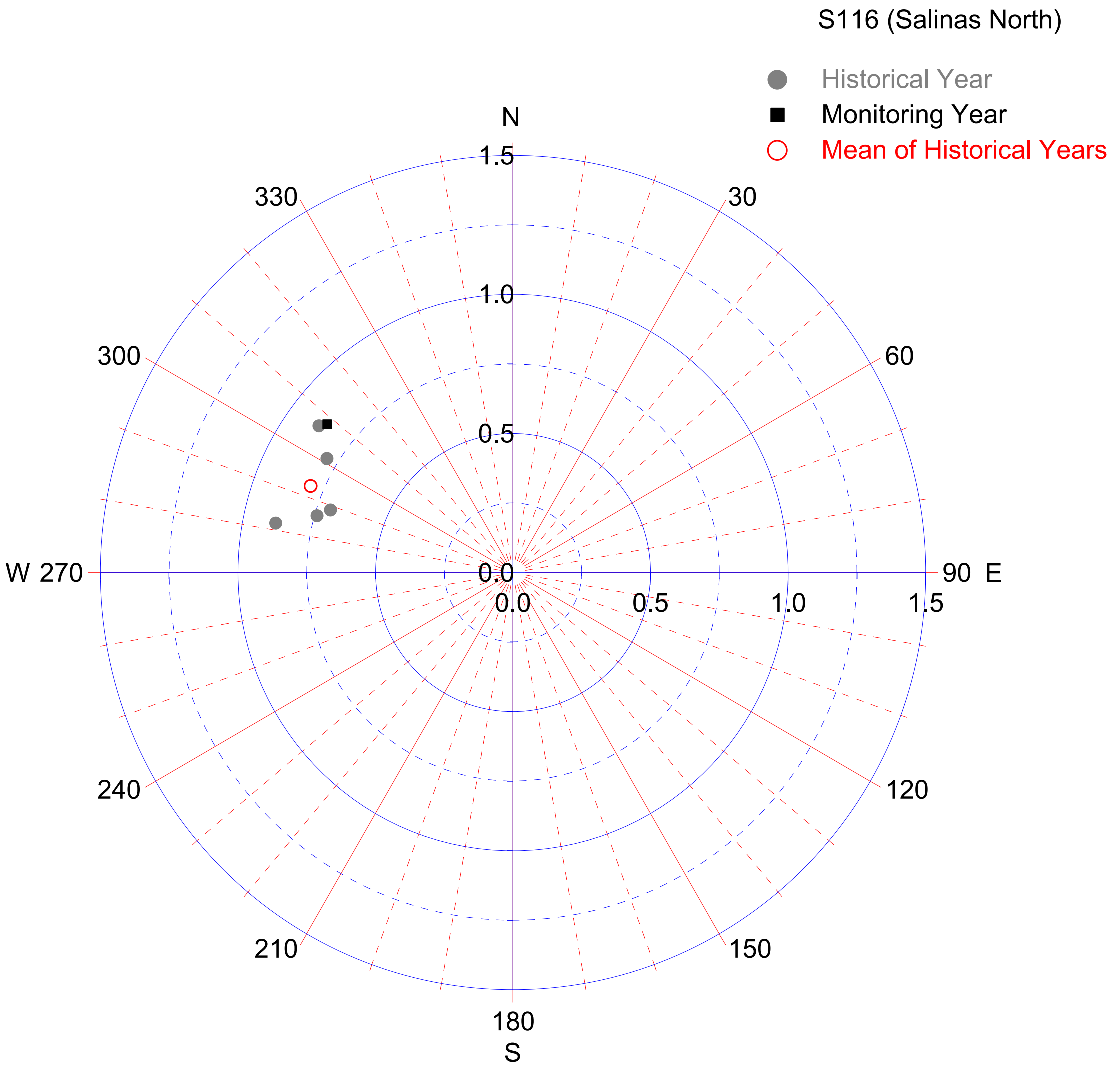

Figure 3b. The average wind speed and direction over the monitoring period (S116). The wind direction was from the symbol to the center, and the wind speed was represented by the distance from the symbol to the center

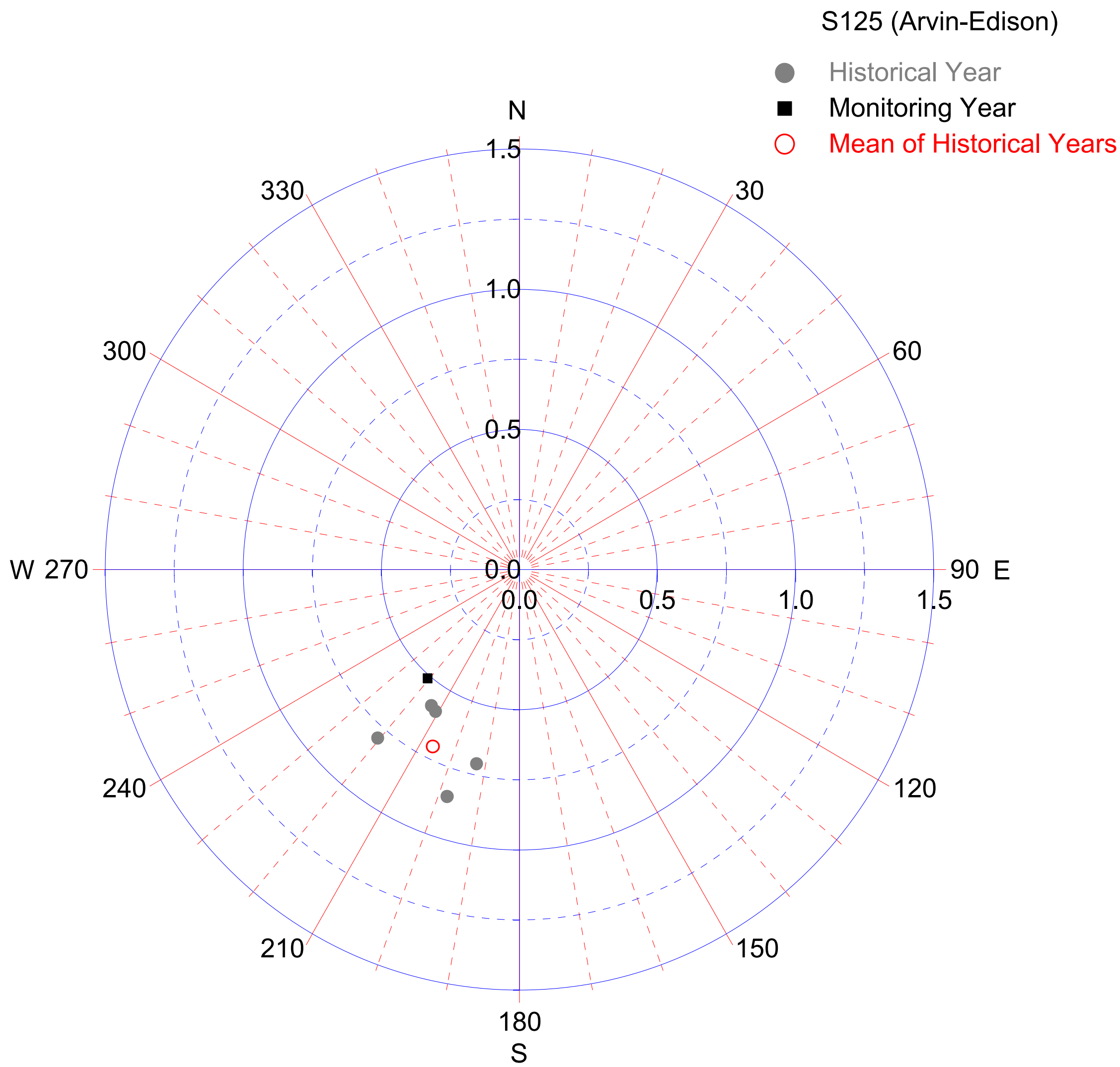

Figure 3c. The average wind speed and direction over the monitoring period (S125). The wind direction was from the symbol to the center, and the wind speed was represented by the distance from the symbol to the center

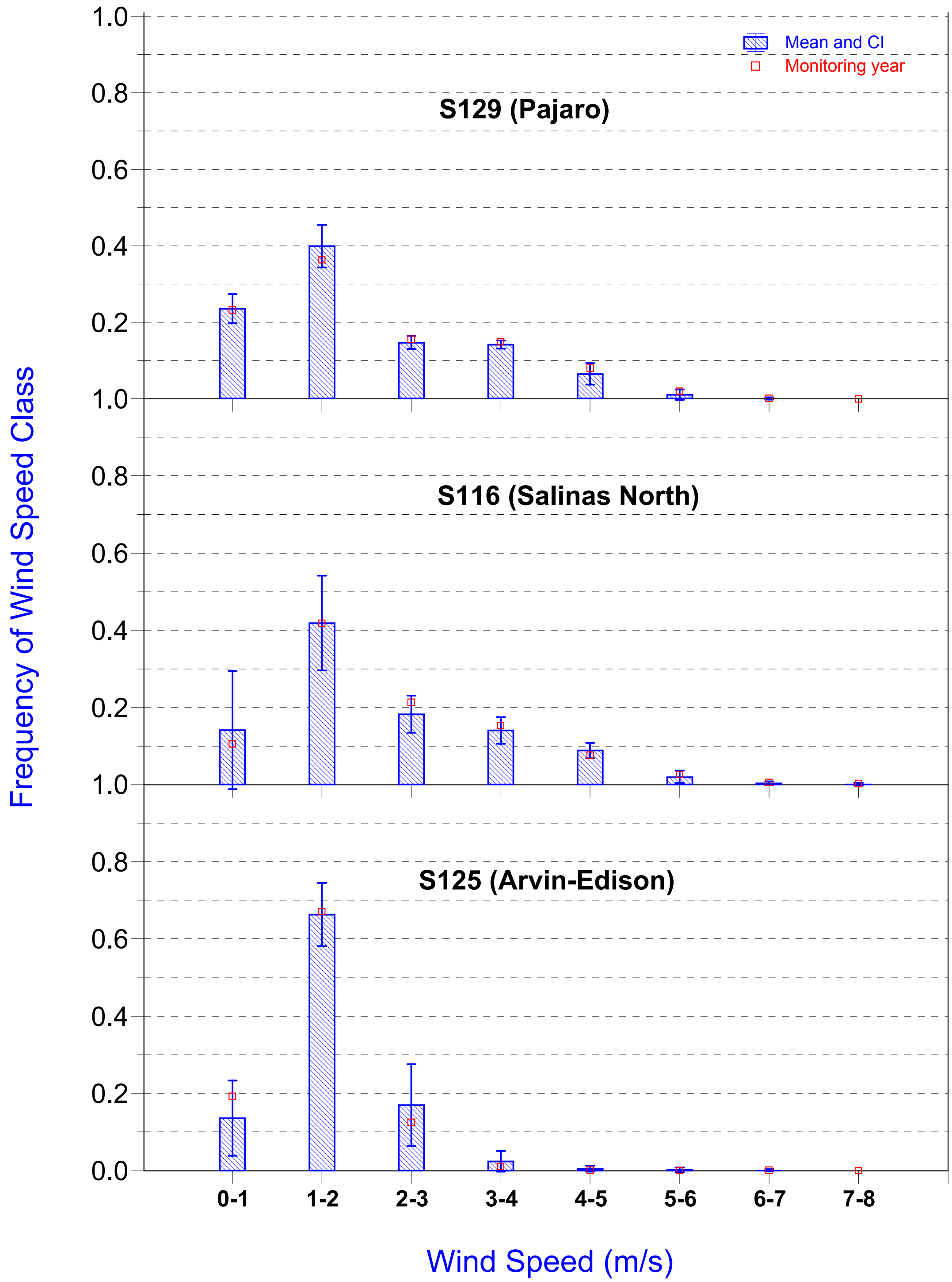

Figure 4. Frequency of wind speed classes for CIMIS weather stations 129, 116 and 125

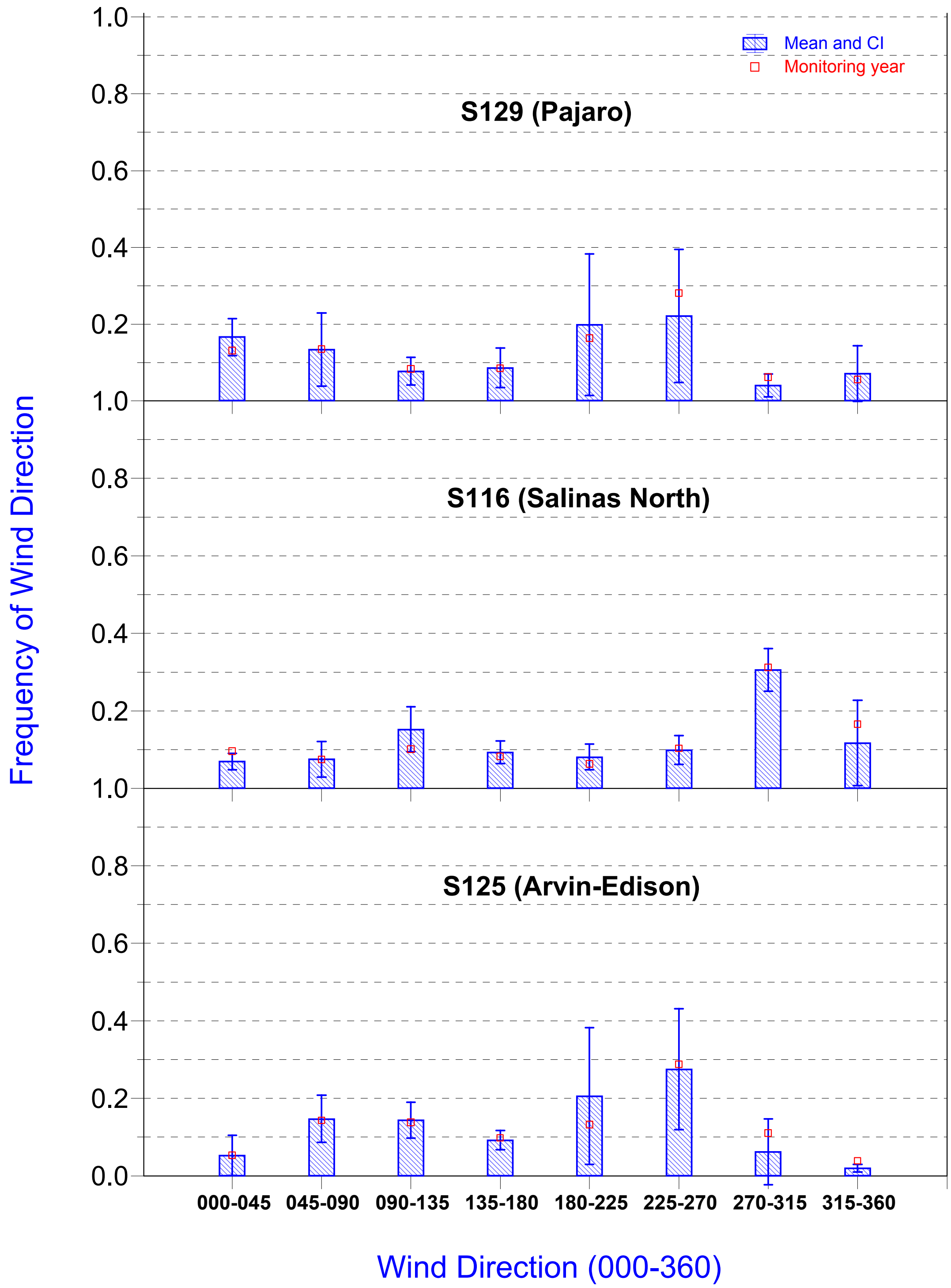

Figure 5. Frequency of wind direction classes for CIMIS weather stations 129, 116 and 125

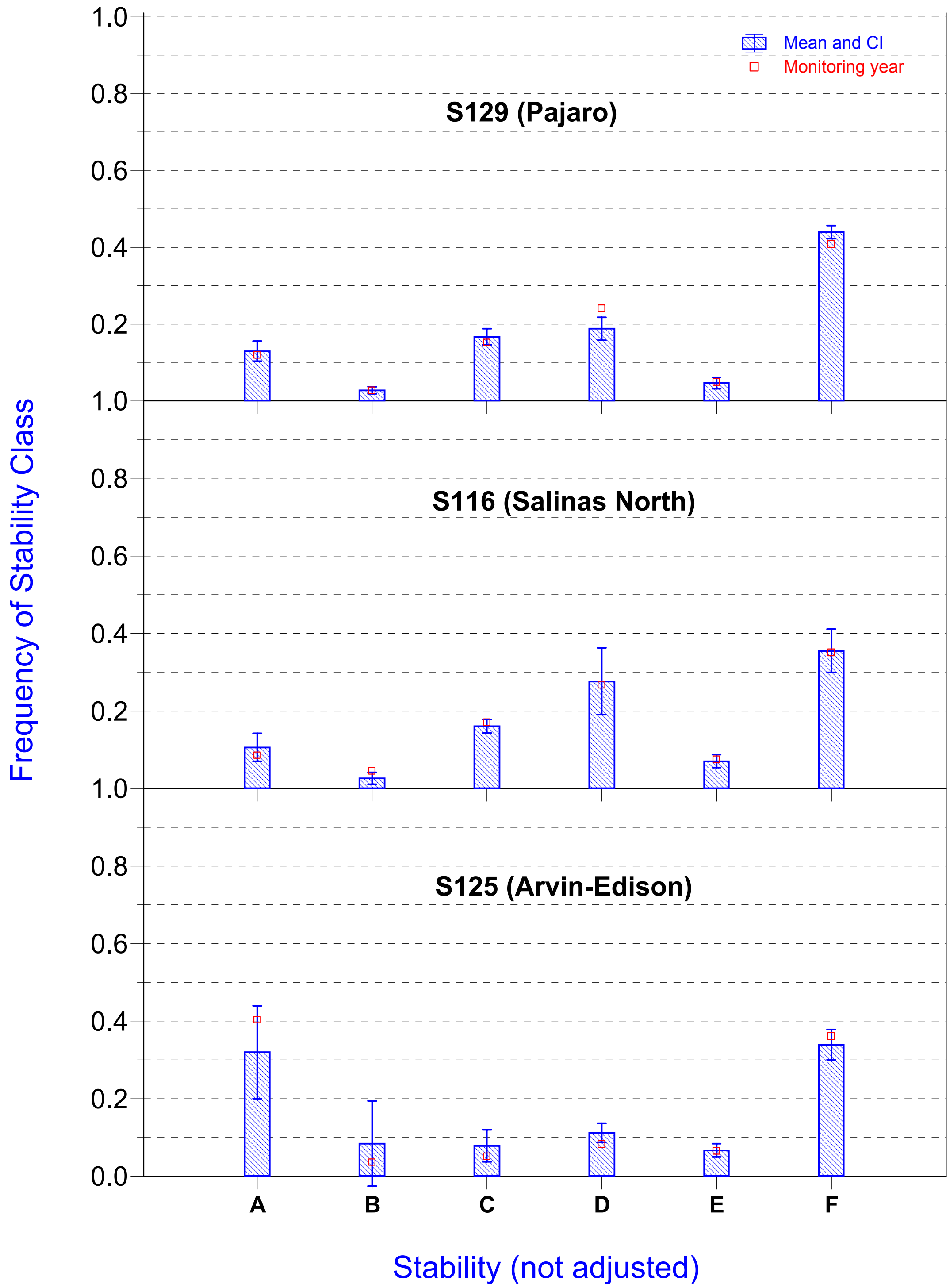

Figure 6. Frequency of stability classes for CIMIS weather stations 129, 116 and 125

Appendix-I

A perl program to calculate Euclidian distance between two sets of locations

```
#!/usr/local/bin/perl  -w
#    Last change:  LI   17 Jul 2001    3:16 pm

# Euclidian.pl
# This program calculates the Euclidian distances from each of the monitoring sites
# to each of the cimis weather stations.

$dir = 'E:\ARB\1807\cimis';
chdir $dir or die "could not find directory $dir\n";

#$Weather_Stations = "Cimis_stations_info.txt";
$Weather_Stations = "Cimis_stations_mont.dat";
$Monitoring_Sites = "Monitoring_sites_mont.dat";
$Euclidian = "Euclidian_Dist_mont.dat";


$starting_year = 95;   # the starting year of the cimis weather station must be operating
                       # in 1995 or earlier
$max_distance = 50;  # the distance between a monitoring site and a weather station
                       # must be 50 km or less


open IN1, "$Weather_Stations" or die "file $Weather_Stations does not exist!";
open OUT, ">$Euclidian"  or die "could not open file $Euclidian !";

print OUT "Station SES     PMS     LJE     SAI     CHU     OAS     Total\n";
L1: while ( $line_IN1 = <IN1>)
{
   chomp $line_IN1;
   ($station_id, $lat_station, $long_station, $elevation,$starting_date) = split(/\t/,$line_IN1);
   #if ($station_id=='station' or $station_id=='') {next L1;}
   ($day, $month, $year) = split(/-/,$starting_date);

   if ($year>$starting_year) {next L1;}
   print OUT "$station_id\t";
   $total=0;

  open IN2, "$Monitoring_Sites" or die "file $Monitoring_Sites does not exist!";
  L2: while ( $line_IN2 = <IN2>)
   {
      chomp $line_IN2;
```

```
        ($site_id, $lat_site, $long_site) = split(/\t/,$line_IN2);
        #print "$site_id, $lat_site, $long_site\n";
        ($lat1, $lat2) = split(/o/,$lat_site);
        ($long1, $long2) =  split(/o/,$long_site);
        $lat2 =~ s/'//;
        $long2 =~ s/'//;;

        $lat_site = $lat1 + $lat2/60;
        $long_site = $long1 + $long2/60;
        $lat_site = int(100*$lat_site +.5)/100;
        $long_site = int($long_site *100 +.5)/100;

        #print "$lat1, $lat2, $lat_site, $long1, $long2, $long_site\n";
        print "$site_id, $lat_site, $long_site\n";

        $unit_factor =2*3.1415926*6370/360; # km/degree


        $X = abs($long_station*cos($lat_station*2*3.1415926/360) -
$long_site*cos($lat_site*2*3.1415926/360));            # this is only true for the same direction, or
small area

        $Y = abs($lat_station  - $lat_site);

        $distance = $unit_factor*sqrt($X*$X + $Y*$Y);
        $distance = int(10*$distance +.5)/10;

#       print "$long_station,$long_site $X,$lat_station,$lat_site, $Y, $distance\n";

        $total += $distance;
        print OUT "$distance\t";

    }
    $total =  int(100*$total+.5)/100;
    print OUT "$total\n";
    close IN2;

}

   close OUT;
```

Appendix-II

A perl program For CIMIS weather data analysis

```
#!/usr/local/bin/perl -w

# Last change:  LI   13 Jul 2001    2:01 pm

# cimis04.pl, a tool for cimis weather data analysis
# added a subroutine for matrix processing on cimis01.pl
# modified the way to calculate the average wind speed and wind direction on cimis02.pl
# also, added using  matrix subroutine to calculate means, STD and STE for cimis elements
# added the confidence intervals (CI) in matrix sub

# use Stability_sigmaA qw(:DEFAULT);

package main;

{
$working_dir = 'E:\ARB\1807\cimis\data';
chdir $working_dir or die "couldn't find the path $working_dir\n";
$station='s125';
subs::sub_mean_stdErr($station);
} #end of package main



package subs;
{

# sub_mean_stdErr
sub sub_mean_stdErr
{
 my($file) = @_;

 $infile = $file.'.dat';
 $outfile = $file.'_out.dat';
 open IN, "$infile" or die "could not open the $infile\n" ;
 open OUT, ">$outfile";

 if (($file eq 's125') or ($file eq 's005')){ # Kern
     $start_mo = 7;
     $start_day= 19;
     $end_mo = 8;
     $end_day = 31;
```

```
    $t_sunrise = 5;
    $t_sunset = 19;}

 else{ # Monterey/Santa Cruz
    $start_mo = 9;
    $start_day= 11;
    $end_mo = 11;
    $end_day = 2;
    $t_sunrise = 6;
    $t_sunset = 18;}


 #initialing ...
   for ( $i=1;$i<=6;$i++ ) {
   $h[$i] = 0; # counter for all records, valid and invalid during the monitoring period, reset for
each year
    for ( $j=1;$j<=17;$j++){
        $n[$i][$j]=0;  # the number of valid records in year 95, 96...99, 00
        $nn[$i][$j]=0; # the number of invalid records in year 95, 96...99, 00
    }
   }

  @element = (
    "00-weather_elements",
    "01-Solar_Radiation(Wm/-2)",
    "02-Net_Radiation(Wm/-2)",
    "03-Soil_Temp_15cm(C)",
    "04-Air_Temp(C)",
    "05-Vapor_Pressure(kPa)",
    "06-Experiment_1",
    "07-Experiment_2",
    "08-Wind_Speed(ms/-1)",
    "09-Resultant_Wind",
    "10-Wind_Direction(0-360)",
    "11-STD._Wind_Dir",
    "12-Precipitation(mm)",
    "13-Not_Used",
    "14-Wind_Gust",
    "15-Hourly_ETo(mm)",
    "16-Relative_H(%)",
    "17-Dew_Point(C)");


 @year = (
    "year",
    "1995",
    "1996",
    "1997",
```

```
   "1998",
   "1999",
   "2000");


 L1: while ($line_IN = <IN>) {

   chomp $line_IN;
   @a =  split(/\,/,$line_IN);
   ($yy,$mm,$dd) = split (/-/,$a[1]);


   if ( ($yy ==0 || (($yy>=95 && $yy<=99))) && (($mm==$start_mo && $dd>=$start_day) ||
($mm>$start_mo && $mm<$end_mo) || ($mm==$end_mo && $dd<=$end_day)) )
      {
      # Determine the $i value, representing the year
      if ($yy==0) {$i=6;}
      else {$i=$yy-94;}

      $h[$i]++;

      for ($j=1;$j<=17;$j++ )
         {
         $xx[$i][$j][$h[$i]] = $a[3+2*$j]; # @xx contains all data (valid and invalid)for the
monitoring period
         $cc[$i][$j][$h[$i]] = $a[2+2*$j]; # @cc contains all comments (valid and invalid) on data
for the monitoring period
         }

      # loop through each of sensors, and count the number of valid records
      for ( $j=1;$j<=17;$j++)    # $j, the sensor serial number
         {
         if ( ($a[3+2*$j]=~ m/^\-|\d/) && ($a[2+2*$j] eq '*') ) { # $j is the sensor number, 3+2*$j is
the column #
            $k = $n[$i][$j]+1; # representing each hour during the mornitoring period
            $x[$i][$j][$k] = $a[3+2*$j]; # @x contains only valid data
            $sum[$i][$j] += $x[$i][$j][$k];
            $n[$i][$j] += 1;
            }
         else
            {$nn[$i][$j] += 1;}

         } # end of for loop

      if (($a[3+2*8]=~ m/^\-|\d/) && ($a[2+2*8] eq '*') && ($a[3+2*10]=~ m/^\-|\d/) && ($a[2+2*10]
eq '*'))
         {
            $sum_WindS_x[$i] += $a[3+2*8]*sin(3.14*$a[3+2*10]/180); # sin and cos take angles
```

```
in radians
            $sum_WindS_y[$i] += $a[3+2*8]*cos(3.14*$a[3+2*10]/180);
         }
      }
   else
     {next L1;}

 } #end of L1 loop

$num_rec = $h;

close IN;



# calculate the mean values for each year
for ( $i=1;$i<=6;$i++ )
{
   for ( $j=1;$j<=17 ;$j++ )
      {
      if ($n[$i][$j] !=0)
         {
         $ave[$i][$j] = $sum[$i][$j]/$n[$i][$j];
         $ave[$i][$j]= int($ave[$i][$j]*100+.5)/100;
         }
      else
         {
         $ave[$i][$j]=-999;
         }
      #print "$n[$i][$j]\t";
      }

   $ave_WindS_x[$i] = $sum_WindS_x[$i]/$n[$i][8];
   $ave_WindS_y[$i] = $sum_WindS_y[$i]/$n[$i][8];
   $ave_WindS[$i] = sqrt($ave_WindS_x[$i]**2+$ave_WindS_y[$i]**2);
   $ave_WindD[$i] = 180*atan2($ave_WindS_x[$i],$ave_WindS_y[$i])/3.14; # atan2 returns a
value in radians, between -pi to pi

   # make necessary adjustment on wind direction

   if ( ($ave_WindS_x[$i]<0 && $ave_WindS_y[$i]<0) || ($ave_WindS_x[$i]<0 &&
$ave_WindS_y[$i]>0) )
      {$ave_WindD[$i] = 360+$ave_WindD[$i];}
   else
      { $ave_WindD[$i] +=0;}

   $ave_WindS[$i] = int($ave_WindS[$i]*100 +.5)/100;
   $ave_WindD[$i] = int($ave_WindD[$i]*10+.5)/10;
```

```
}

# calculate the mean, standard deviation and standard error of wind speed and
# wind direction for period of 1995-1999
for ( $i=1;$i<=5;$i++ )
{
   $sum_WindS_x += $ave_WindS[$i]*sin(3.14/180*$ave_WindD[$i]);  # sin and cos take
parameter in radians
   $sum_WindS_y += $ave_WindS[$i]*cos(3.14/180*$ave_WindD[$i]);
}
   $ave_WindS_x = $sum_WindS_x/5;
   $ave_WindS_y = $sum_WindS_y/5;
   $ave_WindS  = sqrt($ave_WindS_x**2 + $ave_WindS_y**2);
   $ave_WindD = 180/3.14*atan2($ave_WindS_x, $ave_WindS_y);

   # make necessary adjustment on wind direction
   if ( ($ave_WindS_x<0 && $ave_WindS_y<0) || ($ave_WindS_x<0 && $ave_WindS_y>0) )
       {$ave_WindD = 360 + $ave_WindD;}
   else
       { $ave_WindD +=0;}

   $ave_WindS = int($ave_WindS*100+.5)/100;
   $ave_WindD = int($ave_WindD*10+.5)/10;


# calculate the standard deviation and standard error for period of 1995-1999
for ($j=1;$j<=17 ;$j++ )
{
   # get the sum of 5 years
   $Sum_5year[$j]=0;
   $Num_of_Years[$j]=0;
   $Num_of_invalid_Years[$j]=0;

   for ($i=1;$i<=5;$i++)
      {
      if ($ave[$i][$j] != -999)
         {
         $Sum_5year[$j] += $ave[$i][$j];
         $Num_of_Years[$j] += 1;
         }
      else
         {
         $Num_of_invalid_Years[$j] += 1;
         }
      }

   # get the average of 5 years
      $Ave_5year[$j]=0;
```

```
        if ($Num_of_Years[$j] !=0)
            {
            $Ave_5year[$j] += $Sum_5year[$j]/$Num_of_Years[$j];
            }
        else
            {
            $Ave_5year[$j] = -999;
            }



    # calculate the sum of squares
    $Sum_of_square[$j] = 0;

    for ($i=1;$i<=5;$i++ )
        {
        if ($ave[$i][$j] != -999)
            {
            $Sum_of_square[$j] += ($Ave_5year[$j]-$ave[$i][$j])**2;
            }
        }

    if ($Sum_of_square[$j] == 0 )
    {
        $Sum_of_square[$j] = -999;
    }


    # calculate the standard deviation and standard error
    if ($Sum_of_square[$j] !=-999)
    {
        $std[$j] = sqrt($Sum_of_square[$j]);
        $ste[$j] = $std[$j]/sqrt($Num_of_Years[$j]);
    }
    else
    {
         $std[$j] = -999;
          $ste[$j] = -999;
    }
} # end of $j for loop


# formating the numbers
for ( $j=1;$j<=17 ;$j++ )
{

    if ($Ave_5year[$j] != -999)
    {
```

```
        $Ave_5year[$j] = int($Ave_5year[$j]*100+.5)/100;
    }

    if ($std[$j] != -999)
     {
        $std[$j]= int($std[$j]*100+.5)/100;
     }

    if ($ste[$j] != -999)
     {
        $ste[$j]= int($ste[$j]*100+.5)/100;
     }

    if ( $ave[6][$j] != -999 )
    {
        $ave[6][$j] = int($ave[6][$j]*100+.5)/100;
    }
}

goto LLL;
# printing the result
print "Serial#, Sensor, 5-year, std, ste, monitoring-year\n";
print OUT "Serial# \tSensor \t5-year \tstd \tste \tmonitoring year\n";
for ( $j=1;$j<=17 ;$j++ )
    {
    if ( $j==10)
        {
        print "$j \t$element[$j] \t$ave_WindD \t$std[$j] \t$ste[$j] \t$ave_WindD[6]\n";
        print OUT "$j \t$element[$j] \t$ave_WindD \t$std[$j] \t$ste[$j] \t$ave_WindD[6]\n";
        }
        else
        {
        print  "$j \t$element[$j] \t$Ave_5year[$j] \t$std[$j] \t$ste[$j] \t$ave[6][$j]\n";
        print OUT "$j \t$element[$j] \t$Ave_5year[$j] \t$std[$j] \t$ste[$j] \t$ave[6][$j]\n";
        }
    }


LLL:;

# calculate 17 weather elements in cimis data base

# print the title
    print OUT "serial#\tYear\t";
    for ( $j=1;$j<=17 ;$j++ ){
        print OUT "$element[$j]\t";}
    print OUT "\n";
```

```
# convert index from 1 to 0
   for ( $i=1;$i<=5;$i++ )
      {
         print OUT "$i \t$year[$i]\t";
         for ( $j=1;$j<=17 ;$j++ )
            {
            if($j==10)
               {
               $ave0[$i-1][$j-1] = $ave_WindD[$i];
               print "$ave0[$i-1][$j-1]\t";
               print OUT "$ave0[$i-1][$j-1]\t";
               }   # replace wind direction with that calculated with algorithm
            else
               {
               $ave0[$i-1][$j-1] = $ave[$i][$j];
               print "$ave[$i][$j]\t";
               print OUT "$ave[$i][$j]\t";
               }
            }
         print "\n";
         print OUT "\n";
      }

         print "6 \t2000\t";
         print OUT "6 \t2000\t";
         for ( $j=1;$j<=17 ;$j++ )
         {
         print "$ave[6][$j]\t";
         print OUT "$ave[6][$j]\t";
         }

         print "\n";
         print OUT "\n";

   ($mean_ave0, $std_ave0, $ste_ave0, $CI_ave0) = matrix::matrix(@ave0);

   @mean_ave0 = @{$mean_ave0};
   @std_ave0 = @{$std_ave0};
   @ste_ave0 = @{$ste_ave0};
   @CI_ave0 = @{$CI_ave0};

   # replace wind direction with that calculated with algorithm
   $t_0025=2.776;  # t value for n=5, df=4, alfa = 0.05, two tails
   $mean_ave0[9] = $ave_WindD;
   $std_ave0[9] = 81*sqrt(1-$ave_WindS/$Ave_5year[10]);
   $ste_ave0[9] = $std_ave0[9]/sqrt(5);
   $CI_ave0[9] = $ste_ave0[9]*$t_0025;
```

```
    print OUT "\tmean \t@mean_ave0\n";
    print OUT "\tSTD \t@std_ave0\n";
    print OUT "\tSTE \t@ste_ave0\n";
    print OUT "\tC.I \t@CI_ave0\n";




# calculate the wind speed frequency distribution: 0-1, 1-2, ... , 7-8 (m/s)
print  "wind speed frequency distribution\n";
print  "serial# \tYear \t0-1 \t1-2 \t2-3 \t3-4 \t4-5 \t5-6 \t6-7 \t7-8\n";
print OUT "wind speed frequency distribution\n";
print OUT "serial# \tYear \t0-1 \t1-2 \t2-3 \t3-4 \t4-5 \t5-6 \t6-7 \t7-8\n";
for ( $i=1;$i<=6;$i++ )
{
    print"$i \t$year[$i]\t";
    print OUT "$i \t$year[$i]\t";
    for ( $l=0;$l<=7;$l++ )   # $l represents the number of wind categories
    {
        $n_WindS[$i][$l] = 0;
        $f_WindS[$i][$l] = 0;
        for ($k=1;$k<=$n[$i][8];$k++)
        {
            if ($x[$i][8][$k]>=$l && $x[$i][8][$k]<$l+1){
            $n_WindS[$i][$l]++;}
        }
        $f_WindS[$i][$l] = $n_WindS[$i][$l]/$n[$i][8];
        $f_WindS[$i][$l] = int($f_WindS[$i][$l]*1000+.5)/1000;
        $f_WindS0[$i-1][$l] = $f_WindS[$i][$l];
        print "$f_WindS[$i][$l]\t";
        print OUT "$f_WindS[$i][$l]\t";
    }
    print "\n";
    print OUT "\n";
}

($mean_f_WindS0, $std_f_WindS0, $ste_f_WindS0, $CI_f_WindS0) =
matrix::matrix(@f_WindS0);

@mean_f_WindS0 = @{$mean_f_WindS0};
@std_f_WindS0 = @{$std_f_WindS0};
@ste_f_WindS0 = @{$ste_f_WindS0};
@CI_f_WindS0 = @{$CI_f_WindS0};

print OUT "\t mean\t@mean_f_WindS0\n";
print OUT "\t STD\t@std_f_WindS0\n";
print OUT "\t STE\t@ste_f_WindS0\n";
print OUT "\t C.I\t@CI_f_WindS0\n";
```

```
# Calculate frequencies of wind dirrection in 8 intervals: 0-45, 45-90, ..., 315-360.
print "frequencies of wind dirrection in 8 intervals\n";
print "serial# \tYear \t000-045 \t045-090 \t090-135 \t135-180 \t180-225 \t225-270 \t270-315
\t315-360\n";

print OUT "frequencies of wind dirrection in 8 intervals\n";
print OUT "serial# \tYear \t000-045 \t045-090 \t090-135 \t135-180 \t180-225 \t225-270
\t270-315 \t315-360\n";


for ( $i=1;$i<=6;$i++ )
{
  print"$i \t$year[$i]\t";
  print OUT "$i \t$year[$i]\t";

  if ($n[$i][10]!=0)
  {
     for ( $l=0;$l<=7;$l++ )    # $l represents the number of wind directional categories, 45
degrees each
        {
            $n_WindD[$i][$l] = 0;
            $f_WindD[$i][$l] = 0;

            for ($k=1;$k<=$n[$i][10];$k++)
            {
                if ($x[$i][10][$k]>=$l*45 && $x[$i][10][$k]<($l+1)*45){
                $n_WindD[$i][$l]++;}
            }
            $f_WindD[$i][$l] = $n_WindD[$i][$l]/$n[$i][10];
            $f_WindD[$i][$l] = int($f_WindD[$i][$l]*1000+.5)/1000;
            print "$f_WindD[$i][$l]\t";
            print OUT "$f_WindD[$i][$l]\t";
        }
        print "\n";
        print OUT "\n";
   }

   else
    {
      print "no wind direction records available!\n";
      print OUT "no wind direction records available!\n";

    }

}
```

```
for ($i=1;$i<=5;$i++ ) {
   for ( $l=0;$l<=7;$l++ ) {
      $f_WindD0[$i-1][$l] = $f_WindD[$i][$l];
      }
   }

($mean_f_WindD0, $std_f_WindD0, $ste_f_WindD0, $CI_f_WindD0) =
matrix::matrix(@f_WindD0);

@mean_f_WindD0 = @{$mean_f_WindD0};
@std_f_WindD0 = @{$std_f_WindD0};
@ste_f_WindD0 = @{$ste_f_WindD0};
@CI_f_WindD0 = @{$CI_f_WindD0};

print OUT "\t mean\t@mean_f_WindD0\n";
print OUT "\t STD\t@std_f_WindD0\n";
print OUT "\t STE\t@ste_f_WindD0\n";
print OUT "\t C.I\t@CI_f_WindD0\n";



# calculate the prevailing wind direction and its frequency
print "prevailing wind directions and its_frequency\n";
print "serial# \tYear \tstart \tend \tfrequency\n";

print OUT "prevailing wind directions and its frequency\n";
print OUT "serial# \tYear \tstart \tend \tfrequency\n";


for ( $i=1;$i<=6;$i++ )
{
 $f_max[$i]=0;
 $start_WindD[$i]=0;
 $end_WindD[$i]=0;

 if ($n[$i][10]!=0)
 {
   for ( $l=0;$l<=7;$l++ )     # $l represents the number of wind directional categories, 45
degrees each
      {
         $n_WindD[$i][$l] = 0;
         $f_WindD[$i][$l] = 0;

         for ( $m=0;$m<=44;$m++ )   # loop through each degree in each wind directional
category
         {

            for ($k=1;$k<=$n[$i][10];$k++)
```

```
            {
               if ($x[$i][10][$k]>=$l*45+$m && $x[$i][10][$k]<($l+1)*45+$m){
               $n_WindD[$i][$l][$m]++;}
            }

            $f_WindD[$i][$l][$m] = $n_WindD[$i][$l][$m]/$n[$i][10];

            if ($f_WindD[$i][$l][$m] > $f_max[$i])
            {
              $f_max[$i] = $f_WindD[$i][$l][$m];
              $start_WindD[$i] = $l*45+$m;
              $end_WindD[$i] = ($l+1)*45+$m;
            }
          }
       }
    $f_max[$i] = int( $f_max[$i]*1000+.5)/1000;


    print "$i \t$year[$i] \t$start_WindD[$i] \t$end_WindD[$i] \t$f_max[$i]\n";
    print OUT "$i \t$year[$i] \t$start_WindD[$i] \t$end_WindD[$i] \t$f_max[$i]\n";
  }

 else
   {
    print "$i \t$year[$i] \tno wind direction records available!\n";
    print OUT "$i \t$year[$i] \tno wind direction records available!\n";
   }
}


for ($i=1;$i<=5;$i++ ) {
    $f_WindP0[$i-1][0] = $start_WindD[$i];
    $f_WindP0[$i-1][1] = $end_WindD[$i];
    $f_WindP0[$i-1][2] =$f_max[$i];
   }

($mean_f_WindP0, $std_f_WindP0, $ste_f_WindP0, $CI_f_WindP0) =
matrix::matrix(@f_WindP0);

@mean_f_WindP0 = @{$mean_f_WindP0};
@std_f_WindP0 = @{$std_f_WindP0};
@ste_f_WindP0 = @{$ste_f_WindP0};
@CI_f_WindP0 = @{$CI_f_WindP0};

print OUT "\t mean\t@mean_f_WindP0\n";
print OUT "\t STD\t@std_f_WindP0\n";
print OUT "\t STE\t@ste_f_WindP0\n";
print OUT "\t C.I\t@CI_f_WindP0\n";
```

```
# calculate the frequency for stability classes (1-6)

for ( $i=1;$i<=6 ;$i++ )
{
  $rec[$i]=1;  # to record the number of valid records to calulate stability
  $break = 'false';
  for ( $j=1;$j<=6;$j++) # here $j is the stability category 1-6, inializing the counter of each
stability category
       {
        $n_st[$i][$j]=0;
        $n_ad_st[$i][$j]=0;
       }

  L3:for ($k=1;$k<=$h[$i];$k++)  # $k loop variable representing each hour during the
monitoring period
     {                         # here the original data set is used which contains valid and possibly
invalid records
                               # consolidated data set @x can not ensure the same time stamp for both
wind speed and STD of wind direction
        if ( (($xx[$i][8][$k]=~ m/^\-|\d/) && ($cc[$i][8][$k] eq '*')) && (($xx[$i][11][$k]=~ m/^\-|\d/) &&
($cc[$i][11][$k] eq '*')) )
        {

           # judge the day or night time based on preset sunrise and sunset time.
           #if ($x[$i][2][$k]>=$t_sunrise && $x[$i][2][$k]<=$t_sunset)

           # judge the day or night time based on solar radiation
           if ($xx[$i][2][$k]>0)
           {$time = 'Day';}
           else
           {$time ='Night';}

           $st[$i][$k]= stability::sub_Stability_sigmaA($time,$xx[$i][11][$k],$xx[$i][8][$k]);


         # adjusting the stability so that it won't differ more than 1 class from adjacent hours
         # starting from the 1st hour stability
         if ( $rec[$i] == 1 || $break eq 'true' ){  # adjust the stability class from the 2nd hour, or at
the break point
            $ad_st[$i][$k] = $st[$i][$k];
            }
         else{
              if ($st[$i][$k] > $last_st+1) {
                  $ad_st[$i][$k] = $last_st +1 ;}
              elsif ($st[$i][$k] < $last_st-1) {
```

```
            $ad_st[$i][$k] = $last_st -1 ;}
          else {$ad_st[$i][$k] = $st[$i][$k];}
        }

      $break = 'false';
      $rec[$i]+=1;
      $last_st = $ad_st[$i][$k];



      # count the number of each stability category
      for ( $j=1;$j<=6;$j++) # here $j is the stability category 1-6
        {
          if ($st[$i][$k] == $j) {
            $n_st[$i][$j]++;}
          if ($ad_st[$i][$k] == $j) {
            $n_ad_st[$i][$j]++;}
        }


     #print "$i \t$st[$i][$k] \t$ad_st[$i][$k]\n";


    } # end of if block

    else
    {   $st[$i][$k] = -999;
        $ad_st[$i][$k] = -999;
        $break = 'true';
        next;}
   # print OUT "$i \t$st[$i][$k] \t$ad_st[$i][$k]\n";

   } #end of $k loop

}  #end of $i loop


print "frequencies for stability classes (1-6), -- not adjusted\n";
print "serial# \tYear \tA \tB \tC \tD \tE \tF\n";
print OUT "frequencies for stability classes (1-6), -- not adjusted\n";
print OUT "serial# \tYear \tA \tB \tC \tD \tE \tF\n";


for ( $i=1;$i<=6;$i++ )
{
  print "$i \t$year[$i]";
  print OUT "$i \t$year[$i]";
  for ( $j=1;$j<=6;$j++ )
```

```
    {
        $f_st[$i][$j] = int(1000*($n_st[$i][$j]/$rec[$i]) +.5)/1000;
        print "\t $f_st[$i][$j]";
        print OUT "\t $f_st[$i][$j]";
    }
   print"\n";
   print OUT "\n";
}


for ($i=1;$i<=5;$i++ ) {
    for ( $j=1;$j<=6;$j++ ) {
        $f_st0[$i-1][$j-1] = $f_st[$i][$j];
        }
    }

($mean_f_st0, $std_f_st0, $ste_f_st0, $CI_f_st0) = matrix::matrix(@f_st0);

@mean_f_st0 = @{$mean_f_st0};
@std_f_st0 = @{$std_f_st0};
@ste_f_st0 = @{$ste_f_st0};
@CI_f_st0 = @{$CI_f_st0};

print OUT "\t mean\t@mean_f_st0\n";
print OUT "\t STD\t@std_f_st0\n";
print OUT "\t STE\t@ste_f_st0\n";
print OUT "\t C.I\t@CI_f_st0\n";




print "frequencies for stability classes (1-6), -- adjusted\n";
print "serial# \tYear \tA \tB \tC \tD \tE \tF\n";
print OUT "frequencies for stability classes (1-6), -- adjusted\n";
print OUT "serial# \tYear \tA \tB \tC \tD \tE \tF\n";

for ( $i=1;$i<=6;$i++ )
{
  print "$i \t$year[$i]";
  print OUT "$i \t$year[$i]";
  for ( $j=1;$j<=6;$j++ )
    {
        $f_ad_st[$i][$j] = int(1000*($n_ad_st[$i][$j]/$rec[$i])+.5)/1000;
        print "\t $f_ad_st[$i][$j]";
        $f_ad_st0[$i-1][$j-1] = $f_ad_st[$i][$j];
```

```
        print OUT "\t $f_ad_st[$i][$j]";
    }
   print"\n";
   print OUT "\n";
 }


for ($i=1;$i<=5;$i++ ) {
    for ( $j=1;$j<=6;$j++ ) {
        $f_ad_st0[$i-1][$j-1] = $f_ad_st[$i][$j];
        }
    }

($mean_f_ad_st0, $std_f_ad_st0, $ste_f_ad_st0, $CI_f_ad_st0) = matrix::matrix(@f_ad_st0);

@mean_f_ad_st0 = @{$mean_f_ad_st0};
@std_f_ad_st0= @{$std_f_ad_st0};
@ste_f_ad_st0 = @{$ste_f_ad_st0};
@CI_f_ad_st0 = @{$CI_f_ad_st0};

print OUT "\t mean\t@mean_f_ad_st0\n";
print OUT "\t STD\t@std_f_ad_st0\n";
print OUT "\t STE\t@ste_f_ad_st0\n";
print OUT "\t C.I\t@CI_f_ad_st0\n";


close OUT;

L2: print "end!";

} #end of sub

} # end of subroutines package




######################################################################################
####################################
package matrix;
{

    # This sub takes a matrix as parameter, and calculate the mean, standard deviation and
standard error
    # for each column in the matrix
#
```

```
sub matrix
{
   my @matrix = @_;
   $num_rows = @matrix;      # @matrix is a list of references of arrays
   $num_cols = @{@matrix[0]}; # @matrix[0] is a referrence to the array in the first row
                              # therefore, it should be dereferrenced to get the list in the first row
                              # but, in the scaler context, it returns the number of elements in the first row
   # print "col: $num_cols, row: $num_rows\n";

   my (@col_ave, @std, @ste, @CI);

   @t_0025  =   # t_0.025, df = n-1
 (12.706, 4.303, 3.182, 2.776, 2.571, 2.447, 2.365, 2.306, 2.262, 2.228, 2.201, 2.179, 2.160,
2.145, 2.131,
  2.120, 2.110, 2.101, 2.093, 2.086, 2.080, 2.074, 2.069, 2.064, 2.060, 2.056, 2.052, 2.048,
2.045, 1.96);

   # calculate the sums and number of valid elements in each col
   for ($j = 0;$j<$num_cols;$j++)
   {
      $valid_num[$j] = 0;
      $invalid_num[$j] = 0;
      $col_sum[$j] = 0;


      for ( $i=0;$i<$num_rows;$i++ )
      {

       if ($matrix[$i][$j] != -999 )
         {
         $valid_num[$j]++;
         $col_sum[$j] += $matrix[$i][$j];
         }
       else
         {
         $invalid_num[$j]++;
         }
      }


  if ($valid_num[$j] != 0)
   {
   # calculate the means
     $col_ave[$j] = $col_sum[$j]/$valid_num[$j];

   # Calculate the sum_of_square

      $sum_of_square[$j] = 0;
```

```
        for ( $i=0;$i<$num_rows;$i++ )
        {
           if ($matrix[$i][$j] != -999 )
              {
              $sum_of_square[$j] += ($col_ave[$j] - $matrix[$i][$j])**2;
              }
        }


     # Calculate the standard deviation and standard error

        $std[$j] = sqrt($sum_of_square[$j]);
        $ste[$j] = $std[$j]/sqrt($valid_num[$j]);
        $CI[$j] =  $t_0025[$valid_num[$j]-2]*$ste[$j];

     # formating the values
        $col_ave[$j] = int(1000*$col_ave[$j]+.5)/1000;
        $std[$j] = int(1000*$std[$j] + .5)/1000;
        $ste[$j] = int(1000*$ste[$j] + .5)/1000;
        $CI[$j] = int(1000*$CI[$j]+ .5)/1000;
     }

    else
       {
       $col_ave[$j] = -999;
       $std[$j] = -999;
       $ste[$j] = -999;
       $CI[$j] = -999;
       }
     }


     # print "@col_ave\n";

     return (\@col_ave, \@std, \@ste, \@CI);

  } # end of sub

  } # end of package matrix



##############################################################################
####################################
package stability;
{

# calculate the stability based on the standard deviation of wind direction
```

```
# and then adjusted by wind speed at 10m high

# ---------------------------------------------------------- #
# usage: sub_Stability_sigmaA($Time, $STD_WindDirection, $WindSpeed)
sub sub_Stability_sigmaA {
my ($Time, $STD_WindDirection, $WindSpeed) = @_;

$Stability = sub_Stability($STD_WindDirection);
#print "The unadjusted stability is $Stability\n";

# calulate the wind speed at 10m high (assuming the windspeed is
# observed at 2m high and tehg roughness of ground is 0.10m)
# the method employed here is applied to situations under neutral
# atmospheric stability condition.

$WindSpeed = $WindSpeed*(log(10/0.1)/log(2/0.1));

if ($Time eq 'Day') {
        $Stability_sigmaA = sub_day_wind_adjust($Stability,$WindSpeed);
        }
elsif ($Time eq 'Night') {
        $Stability_sigmaA = sub_night_wind_adjust($Stability,$WindSpeed);
        }
else {$Stability_sigmaA = 'x';}

#print ("The adjusted Stability is $Stability_sigmaA\n");

return $Stability_sigmaA;
}


# ---------------------------------------------------------- #
#usage: sub_Stability ($STD_WindDirection)
sub sub_Stability {
my ($STD_WindDirection) = @_;

        if ($STD_WindDirection >= 22.5) {
                $Stability = 1;}
        elsif (   $STD_WindDirection < 22.5 and $STD_WindDirection>=17.5) {
                $Stability = 2;}
        elsif ($STD_WindDirection < 17.5 and $STD_WindDirection>=12.5) {
                $Stability = 3;}
        elsif ($STD_WindDirection < 12.5 and $STD_WindDirection>=7.5) {
                $Stability = 4;}
        elsif ($STD_WindDirection < 7.5 and $STD_WindDirection>=3.8) {
                $Stability = 5;}
        elsif ($STD_WindDirection < 3.8) {
                $Stability = 6;}
```

```
        else {$Stability = -9;}
return $Stability;
}


# ------------------------------------------------------------ #
# usage: sub_day_wind_adjust(Stability,WindSpeed)
sub sub_day_wind_adjust {
my ($Stability,$WindSpeed) = @_;
        if ($Stability == 1) {
                if ($WindSpeed<3) {$ad_Stability = 1;}
                elsif ($WindSpeed>=3 and $WindSpeed<4) {$ad_Stability = 2;}
                elsif ($WindSpeed>=4 and $WindSpeed<6) {$ad_Stability = 3;}
                elsif ($WindSpeed>=6) {$ad_Stability = 4;}
                }
        elsif ($Stability == 2){
                if ($WindSpeed<4) {$Stability = 2;}
                elsif ($WindSpeed>=4 and $WindSpeed<6) {$ad_Stability = 3;}
                elsif ($WindSpeed>=6) {$ad_Stability = 4;}
                }
        elsif ($Stability == 3){
                if ($WindSpeed<6) {$ad_Stability = 3;}
                elsif ($WindSpeed>=6) {$ad_Stability = 4;}
                }
        elsif ($Stability >= 4) {$ad_Stability = 4;}
        else {$ad_Stability = -9;}
return $ad_Stability;
}


# ------------------------------------------------------------ #
#usage: sub_night_wind_adjust(Stability,WindSpeed)
sub sub_night_wind_adjust {
my ($Stability,$WindSpeed) = @_;
        if ($Stability == 1) {
                if ($WindSpeed<2.9) {$ad_Stability = 6;}
                elsif ($WindSpeed>=2.9 and $WindSpeed<3.6) {$ad_Stability = 5;}
                elsif ($WindSpeed>=3.6 and $WindSpeed<6) {$ad_Stability = 4;}
                }
        elsif ($Stability == 2) {
                if ($WindSpeed<2.4) {$ad_Stability = 6;}
                elsif ($WindSpeed>=2.4 and $WindSpeed<3.0) {$ad_Stability = 5;}
                elsif ($WindSpeed>=3) {$ad_Stability = 4;}
                }
        elsif ($Stability == 3) {
                if ($WindSpeed<2.4) {$ad_Stability = 5;}
                elsif ($WindSpeed>=2.4) {$ad_Stability = 4;}
                }
```

```
        elsif ($Stability == 4) {$ad_Stability = 4;}
        elsif ($Stability == 5) {
                if ($WindSpeed<5) {$ad_Stability = 5;}
                elsif ($WindSpeed>=5) {$ad_Stability = 4;}
                }
        elsif ($Stability == 6) {
                if ($WindSpeed<3) {$ad_Stability = 6;}
                elsif ($WindSpeed>=3 and $WindSpeed<5) {$ad_Stability = 5;}
                elsif ($WindSpeed>=5) {$ad_Stability = 4;}
                }
        else {$ad_Stability = -9;}
return $ad_Stability;
}


} # end of package stability
```